

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

ČÍSLICOVÁ SIMULACE KYTAROVÝCH HUDEBNÍCH EFEKTŮ V
REÁLNÉM ČASE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MICHAL PLAČKO

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

ČÍSLICOVÁ SIMULACE KYTAROVÝCH HUDEBNÍCH EFEKTŮ V REÁLNÉM ČASE

REAL-TIME DIGITAL SIMULATION OF GUITAR AUDIO EFFECTS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MICHAL PLAČKO

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JAROMÍR MAČÁK

BRNO 2011



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Michal Plačko

ID: 115257

Ročník: 3

Akademický rok: 2010/2011

NÁZEV TÉMATU:

Číslicová simulace kytarových hudebních efektů v reálném čase

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte metody simulace kytarových hudebních efektů v reálném čase. Zaměřte se hlavně na metodu stavového prostoru se statickou nelinearitou. Provedte simulaci vybraných hudebních efektů v prostředí Matlab a dosažené výsledky porovnejte se simulacemi v prostředí PSpice nebo Microcap. Odladěné algoritmy implementujte v jazyce C++ a optimalizujte pro práci v reálném čase.

DOPORUČENÁ LITERATURA:

- [1] ZÖLZER, U. Digital Audio Signal Processing, 1st ed. New York: McGraw-Hill, Inc., 1997, 290 p. ISBN 0-47-197226-6.
- [2] D.T. Yeh, Digital Implementation of Musical Distortion Circuits by Analysis and Simulation, Disertační práce, Stanford University, June 2009.
- [3] MAČÁK, J.; SCHIMMEL, J. Real- Time Guitar Tube Amplifier Simulation Using an Approximation of Differential Equations. In Proceedings of the 13th International Conference on Digital Audio Effects DAFx10. 2010. s. 34-41. ISBN: 978-3-200-01940- 9.

Termín zadání: 7.2.2011

Termín odevzdání: 2.6.2011

Vedoucí práce: Ing. Jaromír Mačák

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Cieľom práce je oboznámiť sa so spôsobmi simulácie gitarových hudobných efektov, konkrétne s metódou stavového priestoru so statickou nelinearitou, v prostredí Matlab navrhnuť a odladiť algoritmus pre túto metódu a následne odsimulovať vybrané obvody. Odladené algoritmy následne implementovať v jazyku C++ a prispôsobiť na prácu v reálnom čase

KLÚČOVÉ SLOVÁ

Effekt distortion, Číslicová simulácia, Metóda stavového priestoru so statickou nelinearitou (SSMN), Matlab, Diódový orezávač

ABSTRACT

The aim of this work is to study different methods for simulation guitar effects, especially SSMN method. Then write a correctly algorithm for this method in Matlab and apply it on different circuits. Correctly algotirhm implemented in the C++ and adapt to work in real-time

KEYWORDS

Effect distortion, Digital simulation, State space memoryless nonlinearity (SSMN), Matlab, Diode clipper

PLAČKO, Michal *Číslicová simulace kytarových hudebních efektů v reálném čase*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2011. 42 s. Vedúci práce bol Ing. Jaromír Mačák,

PREHLÁSENIE

Prehlasujem, že svoju bakalársku prácu na tému „Číslicová simulace kytarových hudebních efektů v reálném čase“ som vypracoval samostatne pod vedením vedúceho bakalárskej práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, predovšetkým som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomí následkov porušenia ustanovení § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovení § 152 trestného zákona č. 140/1961 Sb.

Brno

.....

(podpis autora)

Pod'akovanie

Ďakujem vedúcemu bakalárskej práce Ing. Jaromírovi Mačákovi za metodickú, pedagogickú a odbornú pomoc a ďalšie cenné rady, ktoré som využil pri tvorbe mojej bakalárskej práce.

Brno

.....

(podpis autora)

OBSAH

Úvod	10
1 Hudobný efekt typu distortion	11
2 Simulácia analógových obvodov	12
2.1 Bilineárna transformácia	12
2.2 Simulácia lineárnych obvodov	13
2.3 Simulácia s využitím numerického riešenia diferenciálnych rovníc . . .	13
2.4 Wave Digital Filtre	13
2.5 Riešenie pomocou aproximácie diferenciálnych rovníc	14
2.6 Metóda stavového priestoru so statickou nelinearitou	15
3 Simulácia distortion efektu	16
3.1 Náhradné zapojenia	17
3.1.1 Náhradné zapojenie tranzistora	17
3.1.2 Náhradné zapojenie diódového orezávača	18
3.2 Diódový orezávač v zapojení s RC filtrom	18
3.3 Diódový orezávač v zapojení s dvoma kondenzátormi	20
3.4 Tranzistorový zosilňovač v zapojení so spoločným emitorom a spätnou vzbudou	22
3.5 Vstupný emitorový sledovač	24
3.6 Zosilňovač s operačným zosilňovačom	25
3.7 Tónová clona distortion efektu	26
3.8 Výstupný emitorový sledovač	27
4 Implementácia algoritmu	28
4.1 Implementácia v prostredí Matlab	28
4.1.1 Funkcia efekt	29
4.1.2 Skripty clipper a clipper2	29
4.1.3 Číslkové filtre	29
4.2 Implementácia v jazyku C++	30
4.2.1 Trieda matica	30
4.2.2 Trieda core	31
4.2.3 Knížnica fcie.h	33
4.3 Implementácia ako VST plug-in modul	33
4.4 Výpočtová náročnosť	34
5 Výsledky	35

6 Záver	39
Literatúra	40
Zoznam symbolov, veličín a skratiek	41

ZOZNAM OBRÁZKOV

1.1	Bloková schéma distortion efektu [10]	11
3.1	Bloková schéma efektu BOSS DS-1 [5]	16
3.2	Náhradná schéma tranzistora [11]	17
3.3	Náhradná schéma zapojenia pre diódový orezávač [9]	18
3.4	RC filter s diódovým orezávačom [10]	19
3.5	Diódový orezávač s dvoma kondenzátormi [11]	20
3.6	Tranzistorový zosilňovačv zapojení SE so spätnou väzbou [11]	22
3.7	Vstupný emitorový sledovač [10]	24
3.8	Modulová frekvenčná charakteristika vstupného emitorového sledovača	24
3.9	Operačný zosilňovač [10]	25
3.10	Modulová frekvenčná charakteristika operačného zosilňovača pre rôzne hodnoty D	25
3.11	Tónová clona distortion efektu [10]	26
3.12	Modulová frekvenčná charakteristika tónovej clony	26
3.13	Výstupný emitorový sledovač distortion efektu	27
3.14	Modulová frekvenčná charakteristika výstupného emitorového sledovača	27
4.1	Vývojový diagram metódy SSMN	28
5.1	Odozva diódového orezávača s RC filtrom - SSMN metóda	35
5.2	Odozva diódového orezávača s dvoma kondenzátormi - SSMN metóda	35
5.3	Odozva tranzistorového zosilňovača so spätnou väzbou - SSMN metóda	35
5.4	Log. spektrogram signálu „Sine sweep“ pre diódový orezávač s RC filtrom	36
5.5	Log. spektrogram signálu „Sine sweep“ pre diódový orezávač s dvoma kondenzátormi	36
5.6	Log. spektrogram signálu „Sine sweep“ pre tranzistorový zosilňovač so spätnou väzbou	37
5.7	Porovnanie simulovaných efektov v Matlabe a Micro Cape	37
5.8	Log. spektrogram signálu „Sine sweep“ distortion efektu - simulácia Micro Cap	38
5.9	Log. spektrogram signálu „Sine sweep“ distortion efektu - simulácia Matlab	38

ÚVOD

Cieľom bakalárskej práce je preštudovať vybrané metódy simulácie, v prostredí Matlab navrhnúť a odladiť algoritmus a následne ho aplikovať na vybrané elektronické obvody.

V hudobnom priemysle má nenahraditeľné miesto analógová technika. Hudobníci nepreferujú digitálne spracovanie signálov, ktoré je označované za sterilné, príliš presné, či chladné. Dávajú prednosť analógovému spracovaniu, práve pre nedokonalosti a nelineárne skreslenia, vďaka ktorým je z tvrdého nepohodlného zvuku príjemný, uchu lahodiaci zvuk. Avšak analógové spracovanie signálov má svoje nevýhody, ktoré sa týkajú analógových obvodov, ako napríklad relatívne krátka životnosť, veľké náklady na údržbu a samozrejme rozmernosť a cena analógových obvodov. To viedlo k číslicovej simulácii analógových obvodov v reálnom čase, takže nie je potrebné kupovať veľké a častokrát drahé zariadenia a zároveň je možný digitálny záznam. Vďaka rýchlemu vývoju výpočtovej techniky bol umožnený vznik nových aplikácií k tomuto účelu. Je teda možné prostredníctvom počítača nasimulovať množstvo analógových obvodov používaných v hudobných štúdiách. Jedným z odvetví je simulácia gitarových hudobných zosilňovačov a efektov, kde je snahou dosiahnuť digitálnu podobu analógového obvodu, vrátane jeho nedokonalostí. Tým sa využije akustická príjemnosť analógového spracovania, spoločne s výhodami digitalizácie, ako napríklad možnosť digitálneho záznamu, nedeštruktívna editácia záznamu, alebo možnosť pristupovať k záznamu na presnosť jednej vzorky.

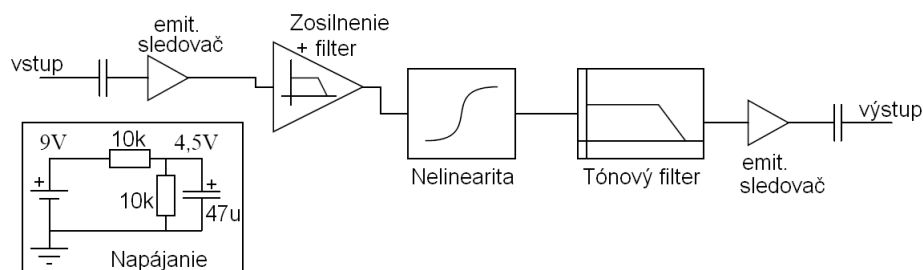
Číslicová simulácia nachádza svoje uplatnenie pri tvorbe multiektov, obsahujúcich simulácie desiatok analógových efektov, či už v domácich, semiprofesionálnych, alebo profesionálnych nahrávacích hudobných štúdiách, kde je možné meniť celé aparatúry od zosilovacích gitarových hláv, cez rôznorodé efekty až k reproxom prakticky bez námahy. Pravdepodobne jediná, no najzávažnejšia nevýhoda číslicovej simulácie je nedostatočne dôveryhodný akustický vnem simulovaných obvodov oproti analógovej predlohe. Avšak zvyšujúcim sa výkonom výpočtovej techniky a vývojom nových algoritmov sa darí túto nevýhodu dostatočne potlačiť.

Z hľadiska aplikácie výpočtových algoritmov je možné rozdeliť obvody na dve skupiny. Obvody, ktoré je možné realizovať algoritmami so statickou prevodovou charakteristikou a obvody, kde je potrebné aplikovať algoritmy založené na dynamickej simulácii nelineárnej prevodovej charakteristiky. Väčšinu obvodov používaných pre realizáciu gitarových hudobných efektov je potrebné realizovať druhou skupinou algoritmov, ktoré majú vysokú výpočtovú náročnosť. Preto sa vyvíjajú stále nové algoritmy, ktorých snahou je výrazne znížiť výpočtovú náročnosť, nie však na úkor vernosti simulácie.

1 HUDOBNÝ EFEKT TYPU DISTORTION

Najrozšírenejšie gitarové hudobné efekty, ako overdrive, distortion, či fuzz, sú založené na rovnakom princípe. Jedná sa o funkčné meniče, ktoré tvarujú signál na základe nelineárnej prevodovej funkcie, napríklad usmernenie, či orezávanie. V hudobnom priemysle sú tieto efekty označované rôznymi názvami (Booster, Super Distortion, Grunge, ...), najčastejšie teda podľa žánru hudby, v ktorom sú najviac používané.

Efekt distortion je nelineárny obvod, ktorý orezáva amplitúdu vstupného signálu, na čo sa všeobecne používajú rôzne formy diódových orezávačov [9], čím v spektre signálu pribúdajú vyššie harmonické zložky a kombinačné spektrálne zložky. Z neskresleného vstupného signálu teda dostaneme signál obohatený o vyššie harmonické zložky, ktoré sú dôvodom nových farieb zvuku. Takto obohatený zvuk je možné upravovať pomocou filtrov a tak dosiahnuť jemne skreslený zvuk, ako aj tvrdé skreslenie, ktoré je základom rockovej muziky.



Obr. 1.1: Bloková schéma distortion efektu [10]

Efekt distortion sa principiálne skladá z niekoľkých blokov, ktorých pospájaním dostaneme výsledné zapojenie. Vďaka tomuto rozdeleniu je možné efekt popísať. Hlavným blokom každého distortion efektu je blok saturačnej nelinearity, ktorý je vsadený medzi filtre. Avšak v konečnom dôsledku je potrebné uvažovať aj skreslenie emitorového sledovača a prvého stupňa zosilnenia [10].

Každý z blokov v blokovej schéme (obr. 1.1) má inú úlohu. Na okrajoch efektu sú emitorové sledovače, ktoré majú za úlohu prispôbovať signál na ďalšie spracovanie. Medzi týmito sledovačmi sú vsadené ďalšie filtre. Ich počet je premenlivý a závisí od výrobcu jednotlivých efektov. Tieto filtre slúžia napríklad na korekcie výsledného signálu (ekvalizér, hlasitosť). Ďalším dôležitým blokom je blok prvotného zosilnenia. Podrobnejší popis všetkých blokov distortion efektu je popísaný v časti (3).

2 SIMULÁCIA ANALÓGOVÝCH OBVODOV

Princíp simulácie spočíva v nadefinovaní funkcií a operácií, ktoré budú pomocou výpočtovej techniky napodobňovať správanie jednotlivých funkčných blokov, ktoré vzniknú zjednodušením analógovej predlohy. Simulované bloky sú následne radené sériovo za seba. Na základe funkcie, ktorú dané bloky spĺňajú, je možné na jednotlivé bloky aplikovať rôzne spôsoby simulácií, využívajúce rôzne algoritmy. V blokových schémach gitarových hudobných efektov sú bloky, ktoré je možné simulovať pomocou algoritmov s lineárnou prevodovou charakteristikou. Medzi tieto bloky patria filtre, ktoré ovplyvňujú úroveň v jednotlivých frekvenčných pásmach. Tieto filtre je možné realizovať pomocou číslicových filtrov, navrhnutých na základe analógovej predlohy [12]. Pomocou algoritmov založených na statickej prevodovej charakteristike je možné dostať sa k dobrým výsledkom pre testovacie signály, najčastejšie harmonický signál rôznej frekvencie a amplitúdy [2]. Avšak sú nepresné pri reálnych obvodoch, ktoré obsahujú akumulčné prvky (cievky, kondenzátory), ktoré môžu spôsobiť posun pracovných bodov, čo sa prejaví väčším výskytom vyšších harmonických zložiek [6]. Tieto zotrvačné prvky môžu meniť tvar prevodovej charakteristiky, ktorý môže byť odlišný pre rôzne frekvencie [2].

2.1 Bilineárna transformácia

Na realizáciu číslicových filtrov podľa analógovej predlohy je potrebné získať koeficienty číslicového filtra z prenosovej funkcie $H(s)$ analógovej predlohy. Vo zvolenom distortion efekte sa vyskytujú len filtre druhého rádu, pre ktoré platí nasledujúci všeobecný tvar prevodovej charakteristiky

$$H(s) = \frac{b_2 s^2 + b_1 s + b_0}{a_2 s^2 + a_1 s + a_0}, \quad (2.1)$$

kde koeficienty $a_0, a_1, a_2, b_0, b_1, b_2$, je možné získať analýzou obvodu, či už ručne, alebo pomocou rôznych simulačných programov, napríklad SNAP. Táto prenosová funkcia však platí pre časovo spojitý signál. K číslicovej simulácii potrebujeme prenosovú funkciu číslicového filtra. K prepočtu na koeficienty číslicového filtra sa využíva bilineárna transformácia, kedy sa substitúciou $s = c \frac{1-z^{-1}}{1+z^{-1}}$ a následnými matematickými úpravami dostaneme k vzťahu

$$H(z) = \frac{B_2 z^{-2} + B_1 z^{-1} + B_0}{A_2 z^{-2} + A_1 z^{-1} + A_0}. \quad (2.2)$$

Táto substitúcia sa prevádza z dôvodu diskretizácie spojitého signálu. Pre koeficienty $A_0, A_1, A_2, B_0, B_1, B_2$, platí nasledovné

$$\begin{aligned} B_0 &= b_0 + b_1c + b_2c^2, \\ B_1 &= 2b_0 - 2b_2c^2, \\ B_2 &= b_0 - b_1c + b_2c^2, \\ A_0 &= a_0 + a_1c + a_2c^2, \\ A_1 &= 2a_0 - 2a_2c^2, \\ A_2 &= a_0 - a_1c + a_2c^2, \end{aligned}$$

kde $c = 2/T$, čo je typická hodnota pre bilineárnu transformáciu [10]. Tieto koeficienty boli využívané pri implementácii číslicových filtrov v jazyku C++.

2.2 Simulácia lineárnych obvodov

Lineárne obvody, teda filtre, môžeme popísať lineárnou prevodovou charakteristikou. Tieto filtre je možné simulovať pomocou číslicových filtrov, ktoré sú navrhnuté na základe analógovej predlohy. Na výpočet koeficientov číslicového filtra sa využíva bilineárna transformácia (2.1).

2.3 Simulácia s využitím numerického riešenia diferenciálnych rovníc

Táto metóda sa v literatúre [9] využíva k simulácii diódového orezávača. Na riešenie diferenciálnych rovníc, ktoré boli zostavené analýzou obvodu, boli použité rôzne metódy, či už z metód prvého rádu (dopredná a spätná Eulerova metóda), alebo z metód druhého a vyššieho rádu (Runge-Kutta metódy a lichobežníkové iteráčné pravidlo). Všetky spomínané metódy pracovali s krokom vzorkovacej periódy a viedli k obdobným výsledkom. Výpočtová náročnosť však bola závislá na počte iterácií a ten bol pre každý testovaný signál odlišný. Pre nelineárne spracovanie signálu v reálnom čase teda tieto metódy nie sú príliš vhodné [2].

2.4 Wave Digital Filtre

Formulácia Wave Digital Filtrov [1] chápe obvod s N prvkami ako N -portovú obvodovú sieť, kde napätie U a prúd I sú nahradené vlnovými premennými A, B , ktoré reprezentujú dopadajúcu a odrazenú vlnu, s impedanciou portu R , ktorá nesmie byť záporná, či rovná nule. Transformované prvky obvodovej schémy (rezistory,

kondenzátory, atď.) sa nachádzajú v [11]. Transformácia vlnových premenných je zadaná následovne

$$A = U + RI, \quad (2.3)$$

$$B = U - RI. \quad (2.4)$$

Prvky sú vzájomne prepojené pomocou tzv. adaptérov [1], ktoré slúžia k prispôbeniu impedancií pripojovaných prvkov. Pri vzájomnom prepojení prvkov je nutné dbať na to, aby v grafoch signálových tokov neboli sľučky bez oneskorovacích blokov, tzv. Delay free loop. Tieto sľučky totiž nie je možné bežne vypočítať. K ich výpočtú sú nutné numerické algoritmy, ktoré však výrazne zvyšujú výpočtovú náročnosť. Aby sa predišlo výskytu týchto sľučiek, každý adaptér obsahuje "reflection-free port", ktorého odrazená vlna nie je závislá na vlne prichádzajúcej, a teda má prispôbenú impedanciu [4]. Nelineárne prvky sú modelované ako závislosť doprednej a odrazenej vlny a impedancia portu závisí na hodnote doprednej vlny. Pokiaľ je nutné simulovať obvod s viacerými nelineárnymi blokmi, je potrebné oddelenie zaradením oneskorovacích blokov, čo však ovplyvňuje presnosť simulácie [4]. Výpočtová náročnosť tejto metódy je pre N prvkov $6N - 12$ operácií násobenia a sčítania [12].

2.5 Riešenie pomocou aproximácie diferenciálnych rovníc

Riešenie pomocou aproximácie nadväzuje na simuláciu s využitím numerického riešenia diferenciálnych rovníc, popísanú v kapitole (2.3). Jedná sa o zostavenie diferenciálnych rovníc pre daný obvod, následnú aplikáciu metódy pre numerické riešenie diferenciálnych rovníc. Môže sa jednať o metódy prvého, druhého, či vyššieho rádu. Výsledky týchto diferenciálnych rovníc sú závislé na vstupných napätiach a stavových premenných systému (napätia kondenzátorov). Je teda možné tento systém považovať za systém s viacerými vstupmi [2]. K zníženiu výpočtovej náročnosti je potrebné prepočítať diferenciálne rovnice a následne ich aproximovať pre rôzne kombinácie vstupných premenných (vstupné napätie, napätia na kondenzátoroch). Aproximované hodnoty môžu byť použité v rámci jednej iterácie pre Newton-Rapsonovu metódu ako odhad výstupného napätia, alebo priamo ako výstupné napätie, ak sú rozostupy medzi vstupným napätím dostatočne malé. Týmto spôsobom sa dosiahne stabilita systému, ak vstupné napätia budú v aproximovaných rozsahoch [2].

2.6 Metóda stavového priestoru so statickou nelinearitou

Nelineárne súčiastky, používané v gitarových efektoch môžu byť popísané charakteristikami bez pamäťového efektu, to znamená, že nie sú závislé na predchádzajúcich stavoch. Obvody teda môžu byť reprezentované v stavovom priestore so separovateľnými podmienkami pre lineárnu a nelineárnu časť. Túto metódu možno označiť ako metódu stavového priestoru so statickou nelinearitou a je odvodená z K-metódy, ktorá sa využíva pri simulácii akustických hudobných systémov [8]. Metódu možno popísať ako

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{C}\mathbf{i}, \quad (2.5)$$

$$\dot{\mathbf{i}} = \mathbf{f}(\mathbf{v}), \quad (2.6)$$

$$\mathbf{v} = \mathbf{D}\mathbf{x} + \mathbf{E}\mathbf{u} + \mathbf{F}\mathbf{i}, \quad (2.7)$$

kde vektor \mathbf{x} je pamäť systému, \mathbf{u} predstavuje vstup do systému a \mathbf{i} je príspevok od nelineárnej časti. Symbol \mathbf{i} bol zvolený zámerne pre nelineárne časti, pretože väčšina nelineárnych prvkov je definovaná ako napätím riadené prúdové zdroje. Nasledne matice \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{D} , \mathbf{E} , \mathbf{F} sú lineárne kombinácie stavov systému, vstupov a nelineárnych častí systému. Vektor \mathbf{v} predstavuje vektor napätí na nelineárnych prvkoch [8]. Nasledujúci výpočtový algoritmus je

1. $\mathbf{p}_n = \mathbf{D}\mathbf{H}(\alpha\mathbf{I} + \mathbf{A})\mathbf{x}_{n-1} + (\mathbf{D}\mathbf{H}\mathbf{B} + \mathbf{E})\mathbf{u}_n + \mathbf{D}\mathbf{H}\mathbf{B}\mathbf{u}_{n-1} + \mathbf{D}\mathbf{H}\mathbf{C}\mathbf{i}_{n-1}$,
2. $\mathbf{i}_n = \mathbf{g}(\mathbf{p}_n)$,
3. $\mathbf{x}_n = \mathbf{H}(\alpha\mathbf{I} + \mathbf{A})\mathbf{x}_{n-1} + \mathbf{H}\mathbf{B}(\mathbf{u}_n + \mathbf{u}_{n-1}) + \mathbf{H}\mathbf{C}(\mathbf{i}_n + \mathbf{i}_{n-1})$,

kde pomocná matica $\mathbf{H} = (\alpha\mathbf{I} - \mathbf{A})^{-1}$, $\alpha = 2/T$, kde T je perióda vzorkovacej frekvencie a $\mathbf{i}_n = \mathbf{g}(\mathbf{p}_n)$ je transformácia definovaná ako

$$\mathbf{i}_n = \mathbf{f}(\mathbf{K}\mathbf{i}_n + \mathbf{p}_n). \quad (2.8)$$

Pre pomocnú maticu \mathbf{K} platí

$$\mathbf{K} = \mathbf{D}\mathbf{H}\mathbf{C} + \mathbf{F}. \quad (2.9)$$

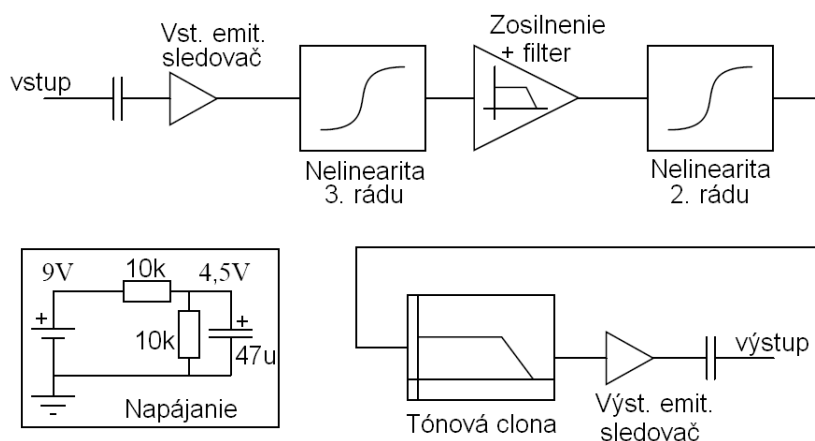
Táto transformácia musí byť riešená numericky, najčastejšie Newtonovou iteračnou metódou [11]. Výstup je následne vyjadrený ako lineárna kombinácia stavu, vstupu a príspevku od nelineárnej časti systému

$$\mathbf{y}_n = \mathbf{A}_o\mathbf{x}_n + \mathbf{B}_o\mathbf{u}_n + \mathbf{C}_o\mathbf{i}_n. \quad (2.10)$$

Ak je táto metóda raz definovaná, stačí zadať matice \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{D} , \mathbf{E} , \mathbf{F} , \mathbf{A}_o , \mathbf{B}_o , \mathbf{C}_o a nelineárnu rovnicu daného obvodu pre popis obvodu a s použitím rovnakého výpočtového algoritmu je možné simulovať viacero rôznych obvodov. Výhodou tejto metódy je, že operácie s maticami môžu byť rýchle [11].

3 SIMULÁCIA DISTORTION EFEKTU

Základom číslicovej simulácie je vhodné rozdelenie analógovej predlohy do blokov. Distortion efekt, ktorý som sa rozhodol simulovať obsahuje dva nelineárne bloky a štyri filtre. Ich usporiadanie môžeme vidieť na blokovej schéme (obr. 3.1). Na základe analýzy jednotlivých blokov je potrebné určiť, akými metódami budú simulované a následne postupovať podľa zvolenej metódy. Pre nelineárne bloky, ktoré boli simulované pomocou metódy stavového priestoru so statickou nelinearitou, bolo potrebné zostaviť obvodové rovnice na popis obvodu pomocou Ohmovho zákona a oboch Kirchhoffových zákonov a upraviť ich na tvar vyhovujúci SSMN metóde. Jedná sa o obvody druhého a tretieho rádu a sú popísané v častiach (3.4, 3.3). Obvod popísaný v časti (3.2) nie je súčasťou simulovaného distortion efektu. Tento obvod slúžil k implementácii a následnému odladeniu algoritmu v prostredí Matlab kvôli jeho jednoduchosti. Jedná sa totiž o obvod prvého rádu. V prípade filtrov, ktoré boli simulované číslicovými filtermi je potrebné aplikovať bilineárnu transformáciu k určeniu koeficientov číslicových filtrov.



Obr. 3.1: Bloková schéma efektu BOSS DS-1 [5]

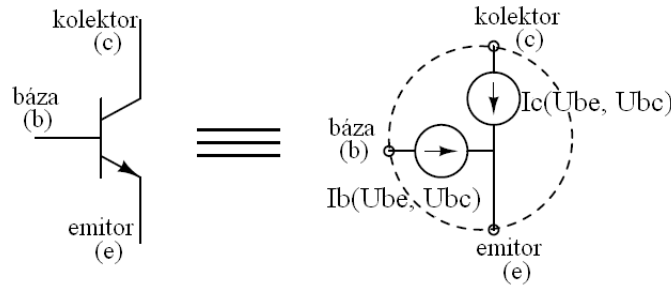
Hlavnými blokmi tohoto distortion efektu sú dva nelineárne bloky. Prvý je umiestnený za emitorovým sledovačom a dochádza tu k zosilneniu signálu a jeho orezaniu. Následne je signál zosilený a privedený na druhý nelineárny blok, kde dôjde k ďalšiemu orezaniu a tvarovaniu signálu. Následne je signál už len upravený tónovou clonou a cez výstupný emitorový sledovač je privedený na výstup.

3.1 Náhradné zapojenia

Hlavné časti nelineárnych blokov tvoria obvody, ktoré je potrebné vhodne nahraďiť prúdovými zdrojmi, ktoré sú riadené napätiami. Vďaka nahradeniu je možné obvody popísať.

3.1.1 Náhradné zapojenie tranzistora

Základná časť prvého nelineárneho obvodu v distortion efekte je tranzistor v zapojení so spoločným emitorom. Tranzistor je možné nahraďiť dvoma prúdovými zdrojmi, zapojenými ako v nasledujúcej schéme (obr. 3.2). Oba zdroje sú prúdové zdroje riadené napätiami. Tranzistor má tri vývody, kolektor, bázu a emitor. Prúdy tečúce do a z týchto vývodov sú riadené napätiami medzi jednotlivými dvojicami týchto vývodov. Napätie $U_{be} = U_b - U_e$ je napätie medzi bázou a emitorom tranzistora a napätie $U_{ce} = U_c - U_e$ je napätie medzi kolektorom a emitorom tranzistora [11]. Všetky parametre nelineárnych súčiastok je možné vyčítať z dokumentácií k jednotlivým súčiastkam.



Obr. 3.2: Náhradná schéma tranzistora [11]

Pre jednoduchosť výpočtov môžeme pre pásmo počuteľných signálov zanedbať nelineárne odpory a parazitné kapacity, ktoré by si vyžadovali modely s podrobnejším popisom. Pre prúdy zjednodušeného modelu tranzistora teda platia nasledujúce rovnice

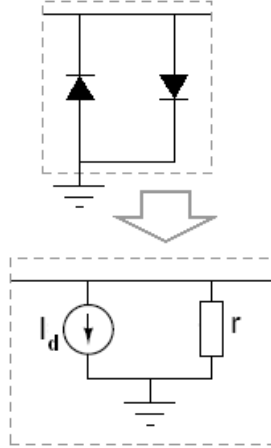
$$I_c = I_s \left(e^{\frac{U_{be}}{U_t}} - 1 \right) - \frac{I_s}{\alpha_R} \left(e^{\frac{U_{bc}}{U_t}} - 1 \right), \quad (3.1)$$

$$I_b = \frac{I_s}{\beta_F} \left(e^{\frac{U_{be}}{U_t}} - 1 \right) + \frac{I_s}{\beta_R} \left(e^{\frac{U_{bc}}{U_t}} - 1 \right), \quad (3.2)$$

kde I_s je saturačný prúd tranzistora, β_R , β_F , α_F sú prúdové zesilňovacie činitele, U_{be} a U_{bc} sú už spomínané napätia medzi jednotlivými vývodmi tranzistora a U_t je termoelektrické napätie tranzistora [11].

3.1.2 Náhradné zapojenie diódového orezávača

V prípade diódového orezávača sa bude jednať len o jeden prúdový zdroj riadený napätím na orezávači. Tento model bude využívaný pri tvorbe obvodových rovníc [9].



Obr. 3.3: Náhradná schéma zapojenia pre diódový orezávač [9]

Diódový orezávač pozostáva z dvoch diód, zapojených paralelne navzájom v opačnom smere. Preto je určite vhodné použiť rovnicu diódy [11]

$$I_d = I_s(e^{\frac{U}{U_t}} - 1). \quad (3.3)$$

Keďže v zapojení sú zväčša dve diódy rovnakého typu, môžeme ich teda pre jednoduchosť považovať za identické. V tom prípade pre model diódového orezávača platí

$$I_d(U) = 2I_s \sinh\left(\frac{U}{U_t}\right), \quad (3.4)$$

kde prúd $I_d(U)$ je prúd pretekajúci oboma diódami, I_s je saturačný prúd diódy, U_t je termálne napätie diódy a napätie U je napätím na diódovom orezávači [11].

3.2 Diódový orezávač v zapojení s RC filtrom

Tento obvod nie je súčasťou simulovaného distortion efektu. Simulovaný bol za účelom oboznámiť sa s SSMN metódou.

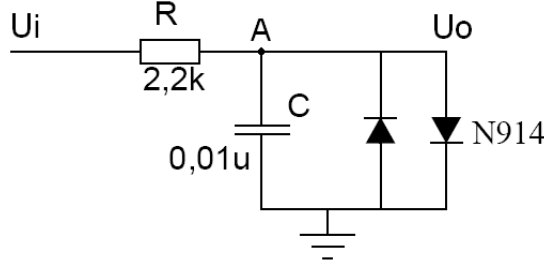
Pre uzol A z obr. 3.4 môžeme podľa 1. Kirchhoffovho zákona napísať nasledovnú obvodovú rovnicu

$$\frac{U_i - U_o}{R} = C \frac{dU_o}{dt} + I_d(U_o). \quad (3.5)$$

z ktorej následnou úpravou dostaneme

$$\frac{dU_o}{dt} = \frac{U_i - U_o}{RC} + \frac{1}{C}I_d(U_o). \quad (3.6)$$

Je zjavné, že sa jedná o diferenciálnu rovnicu prvého rádu s jednou premennou, konkrétne U_o , z čoho hneď možno usúdiť, že $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}, \mathbf{F}, \mathbf{A}_o, \mathbf{B}_o, \mathbf{C}_o$ budú len premenné, nie matice.



Obr. 3.4: RC filter s diódovým orezávačom [10]

Ďalším krokom je úprava diferenciálnej rovnice do podoby rovnice (2.5), voľba stavových premenných, ako aj zostavenie matíc, v tomto prípade však len premen-
ných. Po úprave rovnice (3.6) dostaneme

$$\dot{U}_o = -\frac{G}{C}U_o + \frac{G}{C}U_i - \frac{1}{C}I_d(U_o). \quad (3.7)$$

Ako môžeme vidieť, rovnica (3.7) je analogická s rovnicou (2.5), takže sa nám ju poradilo upraviť správne. Z tejto rovnice sú zjavne odčítateľné stavové premenné $\dot{\mathbf{x}}$, \mathbf{x} , ako aj premenné \mathbf{u} , \mathbf{i} , \mathbf{A} , \mathbf{B} , \mathbf{C} . Keďže nelineárna rovnica je závislá len na jednej premennej, je vďaka rovnici (2.6) môžeme jednoducho určiť aj \mathbf{v}

$$\mathbf{A} = -\frac{G}{C}, \quad \mathbf{B} = \frac{G}{C}, \quad \mathbf{C} = -\frac{1}{C},$$

$$\dot{\mathbf{x}} = \dot{U}_o, \quad \mathbf{x} = U_o, \quad \mathbf{u} = U_i, \quad \mathbf{i} = I_d(U_o), \quad \mathbf{v} = U_o.$$

Obdobným spôsobom sa dostaneme k premenným \mathbf{D} , \mathbf{E} , \mathbf{F} , no najprv je potrebné zapísať rovnicu pre \mathbf{v} , ktorá bude v rovnakom tvare, ako rovnica 2.7

$$U_o = 1 \cdot U_o + 0 \cdot U_i + 0 \cdot I_d(U_o). \quad (3.8)$$

$$\mathbf{D} = 1, \quad \mathbf{E} = 0, \quad \mathbf{F} = 0.$$

Posledné, čo je potrebné k tomu, aby sme mohli pristúpiť k výpočtovému algoritmu, je určenie premenných $\mathbf{A}_o, \mathbf{B}_o, \mathbf{C}_o$. Tieto premenné budú slúžiť k výpočtu

výstupu podľa rovnice (2.10). Keďže napätie na kondenzátore je zároveň napätie výstupné, možno pre výstupné premenné napísať

$$U_o = 1 \cdot U_o + 0 \cdot U_i + 0 \cdot I_d(U_o). \quad (3.9)$$

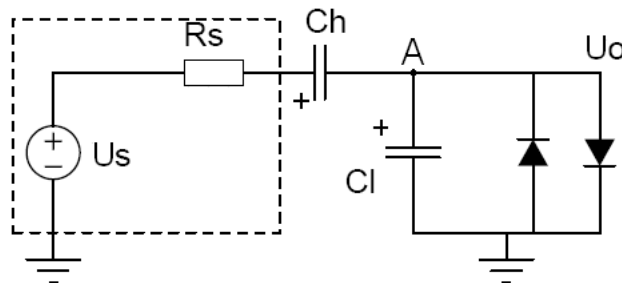
$$A_o = 1, \quad B_o = 0, \quad C_o = 0.$$

Diferenciálna rovnica, ktorú je v algoritme potrebné numericky riešiť, vychádza z rovnice (3.4) pre prúd tečúci diódovým orezávačom. Po úprave tejto rovnice dostaneme rovnicu (3.18).

V tomto momente sú k dispozícii všetky vstupné informácie potrebné pre algoritmus, ktorý vypočíta výstupné hodnoty. Výstupný signál diódového orezávača simulovaného SSMN metódou je zobrazený na (obr. 5.1).

3.3 Diódový orezávač v zapojení s dvoma kondenzátormi

Pre toto zapojenie je rovnako ako v prípade diódového orezávača s RC filtrom nutné dostať sa ku všetkým vstupným hodnotám, potrebným pre výpočtový algoritmus. Postup je analogický predošlému postupu.



Obr. 3.5: Diódový orezávač s dvoma kondenzátormi [11]

Pre uzol A v zapojení (obr. 3.5) spíšeme obvodové rovnice podľa Kirchhoffových zákonov.

$$\frac{U_s - U_x}{R_s} = C_l \frac{dU_o}{dt} + I_d(U_o). \quad (3.10)$$

V tejto rovnici sa nachádza ďalšia neznáma, napätie U_x . Je to napätie medzi rezistorom R_s a kondenzátorom C_h . Pre toto napätie platí

$$U_x = U_o + U_{Ch}. \quad (3.11)$$

Úpravou rovnice (3.10) a dosadením rovnice (3.11) dostaneme rovnicu

$$\dot{U}_o = \frac{G_s}{C_1}(U_s - U_o - U_{Ch}) - \frac{1}{C_1}I_d(U_o). \quad (3.12)$$

Prúd rezistorom R_s je zároveň prúd kondenzátorom C_h a preto je možné napísať

$$\frac{U_s - U_o}{R_s} = C_h \dot{U}_{Ch} \quad (3.13)$$

Dosadením rovnice (3.11) do rovnice (3.13) a následnou úpravou dostaneme rovnicu

$$\dot{U}_{Ch} = \frac{G_s}{C_h}(U_s - U_o - U_{Ch}). \quad (3.14)$$

Rovnice (3.12) a (3.14) sú obvodové rovnice obvodu (obr. 3.5). Jedná sa o sústavu diferenciálnych rovníc s dvoma neznámymi. Úpravou týchto rovníc do tvaru rovnice (2.5) získame rovnicu

$$\begin{bmatrix} \dot{U}_o \\ \dot{U}_{Ch} \end{bmatrix} = \begin{bmatrix} -\frac{G_s}{C_1} & -\frac{G_s}{C_1} \\ -\frac{G_s}{C_h} & -\frac{G_s}{C_h} \end{bmatrix} \cdot \begin{bmatrix} U_o \\ U_{Ch} \end{bmatrix} + \begin{bmatrix} \frac{G_s}{C_1} \\ \frac{G_s}{C_h} \end{bmatrix} \cdot U_s + \begin{bmatrix} -\frac{1}{C_1} \\ 0 \end{bmatrix} \cdot I_d(U_o). \quad (3.15)$$

Z tejto rovnice je možné vypísať nasledujúce hodnoty

$$\mathbf{A} = \begin{bmatrix} -\frac{G_s}{C_1} & -\frac{G_s}{C_1} \\ -\frac{G_s}{C_h} & -\frac{G_s}{C_h} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \frac{G_s}{C_1} \\ \frac{G_s}{C_h} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} -\frac{1}{C_1} \\ 0 \end{bmatrix},$$

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{U}_o \\ \dot{U}_{Ch} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} U_o \\ U_{Ch} \end{bmatrix}, \quad \mathbf{u} = U_s, \quad \mathbf{i} = I_d(U_o), \quad \mathbf{v} = U_o.$$

Ďalej je potrebné zapísať U_o v tvare, v akom je rovnica (2.7)

$$U_o = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{U}_o \\ \dot{U}_{Ch} \end{bmatrix} + 0 \cdot U_s + 0 \cdot I_d(U_o). \quad (3.16)$$

Z tejto rovnice zapíšene premenné \mathbf{D} , \mathbf{E} , \mathbf{F}

$$\mathbf{D} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} 0 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} 0 \end{bmatrix},$$

K výpočtu výstupu sú ešte potrebné premenné \mathbf{A}_o , \mathbf{B}_o , \mathbf{C}_o . Výstupné napätie je, ako v prvom prípade, napätie na diódovom orezávači, teda U_o . Pre toto napätie je možné podľa rovnice (2.10) napísať

$$U_o = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{U}_o \\ \dot{U}_{Ch} \end{bmatrix} + 0 \cdot U_s + 0 \cdot I_d(U_o). \quad (3.17)$$

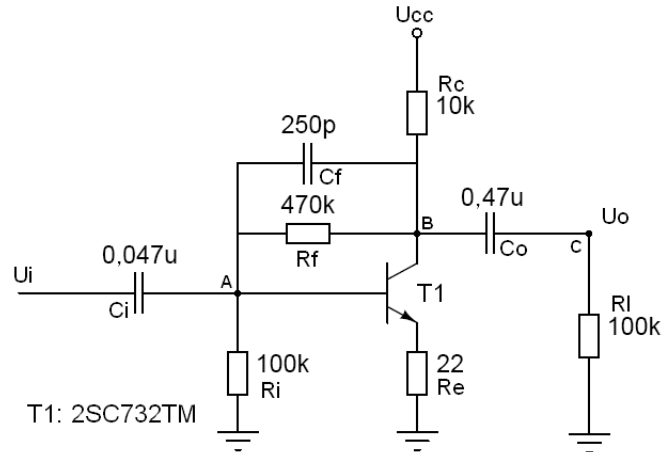
$$\mathbf{A}_o = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \mathbf{B}_o = \begin{bmatrix} 0 \end{bmatrix}, \quad \mathbf{C}_o = \begin{bmatrix} 0 \end{bmatrix}.$$

Nakoľko sa v tomto prípade jedná o rozšírenie obvodu (obr. 3.4) len o jeden kondenzátor, nelineárna časť obvodu, diódový orezávač, ostáva nezmenená. Rovnako teda ostáva nezmenená aj diferenciálna rovnica, ktorá sa bude vo výpočtovom algoritme riešiť

$$\mathbf{0} = 2 \cdot I_s \cdot \sinh\left(\frac{U_o}{U_t}\right) - I_d(U_o). \quad (3.18)$$

3.4 Tranzistorový zosilňovač v zapojení so spoločným emitorom a spätnou väzbou

Pri obvode (obr. 3.6) sa jedná o prvý skresľovací obvod efektu distortion, ktorý je umiestnený hneď za emitorovým sledovačom. Ide o jednostupňový tranzistorový zosilňovač v zapojení so spoločným emitorom so spätnou väzbou. Jadrom tohoto obvodu je samotný tranzistor, ktorý bude pri analýze nahradený modelom, ktorý je popísaný v časti (3.1.1).



Obr. 3.6: Tranzistorový zosilňovačv zapojení SE so spätnou väzbou [11]

Obdobným spôsobom ako pri obvodoch (obr. 3.4, 3.5) je možné zostaviť obvodové rovnice pre body A, B a C na obrázku (obr. 3.6). Rovnice sú zostavené s využitím Ohmovho zákona a Kirchhoffových zákonov. Ich Následnou úpravou dostaneme nasledujúce obvodové rovnice

$$\dot{U}_{Ci} = \frac{G_c + G_l + G_i}{C_i}(U_i - U_{Ci}) - \frac{G_c + G_l}{C_i}(U_{bc}) - \frac{G_l}{C_i}(U_{Co} + U_{cc}) + \frac{1}{C_i}(I_b + I_c), \quad (3.19)$$

$$\dot{U}_{bc} = \frac{G_c + G_l}{C_f}(U_i - U_{Ci}) - \frac{G_c + G_l + G_f}{C_f}(U_{bc}) - \frac{G_l}{C_f}(U_{Co} + U_{cc}) + \frac{1}{C_f}I_c, \quad (3.20)$$

$$\dot{U}_{Co} = -\frac{G_1}{C_o}(U_{Ci} + U_{bc} + U_{Co} - U_i). \quad (3.21)$$

Úpravou obvodových rovníc na tvar rovnice (2.5), ktorá vyhovuje SSMN metóde je možné určiť nasledujúce hodnoty

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} -\frac{G_c+G_l+G_i}{C_i} & -\frac{G_c+G_l}{C_i} & -\frac{G_l}{C_i} \\ -\frac{G_c+G_l}{C_f} & -\frac{G_c+G_l+G_f}{C_f} & -\frac{G_l}{C_f} \\ -\frac{G_l}{C_o} & -\frac{G_l}{C_o} & -\frac{G_l}{C_o} \end{bmatrix}, \\ \mathbf{B} &= \begin{bmatrix} \frac{G_c+G_l+G_i}{C_i} & -\frac{G_c}{C_i} \\ \frac{G_c+G_l}{C_h} & -\frac{G_c}{C_f} \\ \frac{G_l}{C_o} & 0 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \frac{1}{C_i} & \frac{1}{C_i} \\ 0 & \frac{1}{C_o} \\ 0 & 0 \end{bmatrix}, \\ \dot{\mathbf{x}} &= \begin{bmatrix} \dot{U}_{Ci} \\ \dot{U}_{bc} \\ \dot{U}_{Co} \end{bmatrix}, \mathbf{x} = \begin{bmatrix} U_{Ci} \\ U_{bc} \\ U_{Co} \end{bmatrix}, \mathbf{u} = \begin{bmatrix} U_i \\ U_{cc} \end{bmatrix}, \mathbf{i} = \begin{bmatrix} I_b(U_{be}, U_{bc}) \\ I_c(U_{be}, U_{bc}) \end{bmatrix}, \mathbf{v} = \begin{bmatrix} U_{be} \\ U_{bc} \end{bmatrix}. \end{aligned}$$

Ďalej zostavíme rovnice pre vektor $\mathbf{v} = [U_{be} \ U_{bc}]^T$ a ich následnou úpravou na tvar rovnice (2.7) určíme ďalšie premenné, konkrétne \mathbf{D} , \mathbf{E} , \mathbf{F}

$$U_{be} = U_i - U_{Ci} - R_e I_b - R_e I_c \quad (3.22)$$

$$\mathbf{D} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} -R_e & -R_e \\ 0 & 0 \end{bmatrix},$$

Pre určenie premenných \mathbf{A}_o , \mathbf{B}_o , \mathbf{C}_o je potrebné zapísať vzťah pre výstupné napätie, opäť s pomocou Ohmovho zákona. Následne tento vzťah upraviť do tvaru rovnice (2.10) a vyčítať nasledujúce hodnoty

$$\mathbf{A}_o = \begin{bmatrix} -1 & -1 & -1 \end{bmatrix}, \quad \mathbf{B}_o = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \mathbf{C}_o = \begin{bmatrix} 0 & 0 \end{bmatrix}.$$

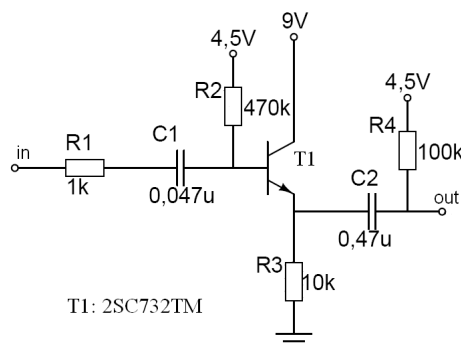
Nakoľko nelineárnou časťou tohoto obvodu je tranzistor, príspevok od nelineárnej časti bude v algoritme reprezentovaný sústavou nelineárnych rovníc pre básový a kolektorový prúd [11]. Táto sústava bude riešená numerickými metódami a je potrebné prepísať ju do vyhovujúceho tvaru, ktorý je

$$\mathbf{0} = I_s \left(e^{\frac{U_{be}}{U_t}} - 1 \right) - \frac{I_s}{\alpha_R} \left(e^{\frac{U_{bc}}{U_t}} - 1 \right) - I_c, \quad (3.23)$$

$$\mathbf{0} = \frac{I_s}{\beta_F} \left(e^{\frac{U_{be}}{U_t}} - 1 \right) + \frac{I_s}{\beta_R} \left(e^{\frac{U_{bc}}{U_t}} - 1 \right) - I_b. \quad (3.24)$$

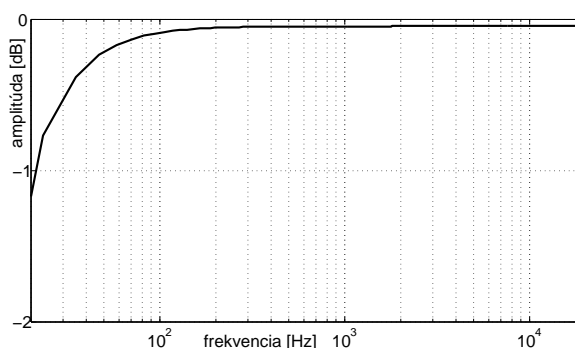
3.5 Vstupný emitorový sledovač

Vstupný emitorový sledovač má za úlohu prispôbienie signálu snímaného z gitarových snímačov. Je to prvý blok efektu distortion a jedná sa o filter druhého rádu.



Obr. 3.7: Vstupný emitorový sledovač [10]

Z nasledujúcej modulovej frekvenčnej charakteristiky je zrejmé, že emitorový sledovač má zosilenie takmer rovné 1. Teda v rámci počuteľných signálov je možné ho



Obr. 3.8: Moduová frekvenčná charakteristika vstupného emitorového sledovača

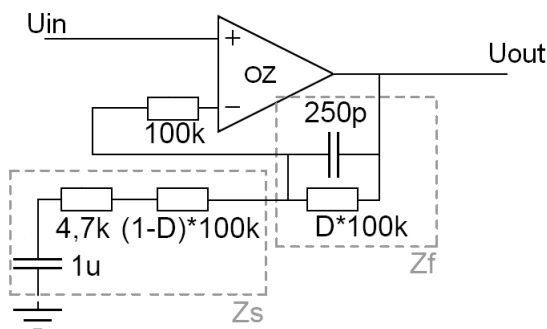
pokladať za lineárny obvod, čo má za následok značné zjednodušenie simulácie. Koeficienty analógovej predlohy (obr. 3.7) boli získané pomocou programu SNAP. Následne bola aplikovaná bilineárna transformácia. Teda koeficienty analógovej predlohy boli prepočítané na koeficienty číslicového filtra pomocou vzťahov, ktoré sú uvedené v časti (2.1).

3.6 Zosilňovač s operačným zosilňovačom

Tento obvod je tvorený operačným zosilňovačom v neinvertujúcom zapojení, ako je znázornený na (obr. 3.9). Napäťové zosilenie tohoto zosilňovača je vyjadrené následovne

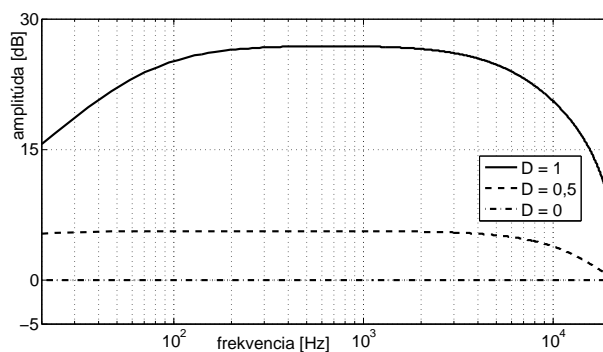
$$\frac{U_o}{U_i} = \frac{Z_s + Z_f}{Z_s}. \quad (3.25)$$

Blok je umiestnený za prvý nelineárny blok a slúži na zosilenie signálu, ktorý je ná-



Obr. 3.9: Operačný zosilňovač [10]

sledne spracovaný ďalším nelineárnym blokom. Pre jednoduchosť simulácie budeme považovať operačný zosilňovač za ideálny. Zosilenie tohoto obvodu je určené ovlá-

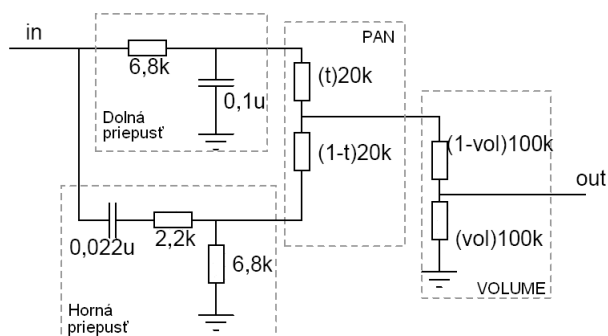


Obr. 3.10: Modulová frekvenčná charakteristika operačného zosilňovača pre rôzne hodnoty D

dacím prvkom „DIST“. Jedná sa o potenciometer zasahujúci do oboch impedancií ovplyvňujúcich výsledný napäťový prenos. V schéme je reprezentovaný rezistormi $(1 - D) \cdot 100k$ a $D \cdot 100k$. D nadobúda hodnoty 0 – 1.

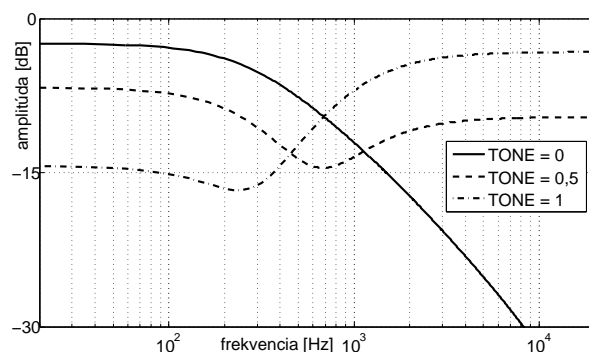
3.7 Tónová clona distortion efektu

Tónová clona je blok, ktorý nasleduje v distortion efekte po druhom bloku nelinearity. Jeho úlohou je upraviť signál pred tým, ako bude privedený na výstupný emitorový sledovač. Ako môžeme vidieť na schéme (obr. 3.14), tento ekvalizér sa



Obr. 3.11: Tónová clona distortion efektu [10]

skladá z dolnej a hornej priepusti. Tieto dva selektívne obvody sú zapojené paralelne a ich výstupy sú pripojené na ovládací prvok „TONE“, ktorý je realizovaný potenciometrom. Z bežca tohoto potenciometra je upravený signál privedený na posledný ovládací prvok „VOL“, ktorý je opäť realizovaný pomocou potenciometra a ovláda hlasitosť výstupného signálu.

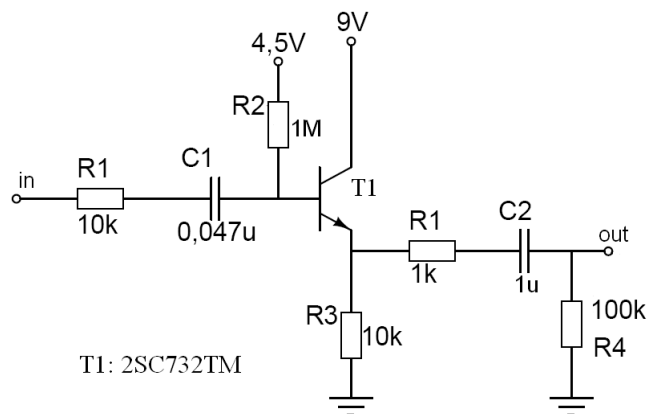


Obr. 3.12: Modulová frekvenčná charakteristika tónovej clony

Ak je potenciometer zapojený ako odporový delič, ako je tomu v prípade potenciometra na ovládanie hlasitosti, je potrebné uvažovať, že napäťový prenos sa mení logaritmicky. Preto musíme prepočítať hodnotu ovládacieho prvku tak, aby pri 50% bola dosiahnutá polovičná hlasitosť.

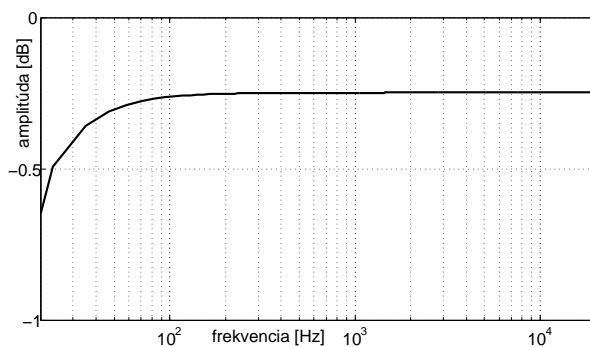
3.8 Výstupný emitorový sledovač

Výstupný emitorový sledovač je posledný blok distortion efektu. Slúži na prispôsobenie signálu na výstup. Jedná sa o obvod veľmi podobný vstupnému emitorovému sledovaču. Modulová frekvenčná charakteristika je podobná ako v prípade vstupného



Obr. 3.13: Výstupný emitorový sledovač distortion efektu

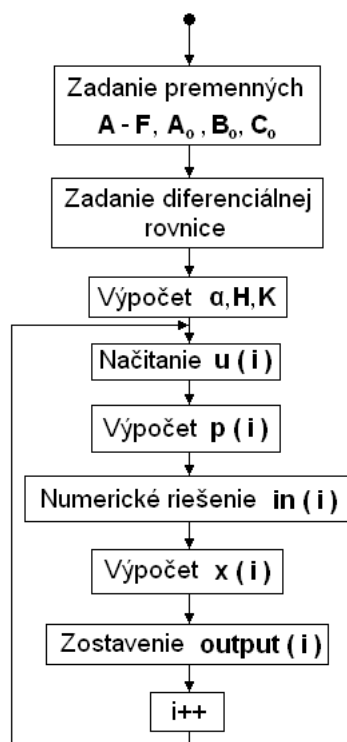
emitorového sledovača, má zosilenie takmer rovné 1. Tento blok môžeme rovnako považovať za lineárny v rámci počuteľných signálov.



Obr. 3.14: Modulová frekvenčná charakteristika výstupného emitorového sledovača

4 IMPLEMENTÁCIA ALGORITMU

Samotnej implementácii výpočtového algoritmu pre metódu stavového priestoru so statickou nelinearitou predchádzalo zostavenie vývojového diagramu (obr. 4.1). Na základe tohoto vývojového diagramu bol zostavený algoritmus ako v prostredí Matlab, tak aj v jazyku C++. Všetky zdrojové kódy, ako z prostredia Matlab, tak aj z jazyka C++ sú v prílohe na CD.



Obr. 4.1: Vývojový diagram metódy SSMN

4.1 Implementácia v prostredí Matlab

Cieľom implementácie v prostredí Matlab bolo vytvoriť nezávislú funkciu simulujúcu efekt distortion, ktorá bude spracovávať signál ľubovoľnej dĺžky, ktorý bude zadáný ako jeden z parametrov funkcie. Ďalšie parametre funkcie slúžia na nastavenie parametrov efektu, ako úroveň skreslenia, nastavenie tónovej clony a nastavenie hlasitosti. Výsledkom funkcie má byť skreslený signál. Výsledkom práce je funkcia **efekt**, ktorá spĺňa tieto požiadavky a podrobnejšie je popísaná v nasledujúcej časti.

4.1.1 Funkcia efekt

Jedná sa o funkciu s jednou výstupnou hodnotou, výstupným skresleným napätím. Všeobecný zápis funkcie v programe je nasledovný

```
out = efekt(dist,tone,vol,fvz,input).
```

Vstupných parametrov je päť. Prvé tri parametre určujú nastavenie ovládacích prvkov efektu distortion, konkrétne nastavenie skreslenia, nastavenie tónovej clony a nastavenie hlasitosti. Štvrtý parameter je vzorkovacia frekvencia, s ktorou bude efekt pracovať a piatym, posledným parametrom je vstupný signál, ktorý má byť spracovaný. Telo funkcie obsahuje volanie jednotlivých skriptov pre číslicové filtre, ako aj skripty obsahujúce nasimulované skreslovače. Poradie jednotlivých skriptov zodpovedá poradiu blokov v blokovej schéme efektu distortion, nakoľko je potrebné dodržať postupnosť operácií. Ďalej tiež obsahuje funkcie na vykreslenie signálu po jednotlivých operáciách a frekvenčné charakteristiky jednotlivých filtrov pomocou funkcií `subplot`, `plot` a `semilogx`.

4.1.2 Skripty clipper a clipper2

Zadávanie premenných je realizované cez samostatné skripty, ktoré je možné obmieniať pre rôzne obvody. Zadávanie nelineárnej rovnice je riešené prostredníctvom funkcie, ktorá obsahuje rovnicu popisujúcu nelineárnu časť simulovaného obvodu. Táto rovnica je upravená na požadovaný tvar. Funkcia je následne volaná v algoritme prostredníctvom funkcie `fsolve`, ktorá aproximuje ku koreňom nelineárnej rovnice. Oba skripty sú v podstate rovnaké. Jediný rozdiel spočíva vo volaní rozdielných skriptov na určenie vstupných premenných a zadávanie inej funkcie pri numerickom riešení diferenciálnej rovnice. Pri tvorbe algoritmu bolo potrebné ošetriť najmä prípad prvej vzorky, keď neboli dostupné žiadne predchádzajúce výsledky, ktoré sú potrebné pri výpočtoch. Došlo teda k zjednodušeniu výpočtového algoritmu, nakoľko boli preschádzajúce hodnoty považované za nulové. Tento stav je vyriešený cez jednoduchú podmienku.

4.1.3 Číslicové filtre

Simulácia v prostredí Matlab obsahuje štyri skripty simulujúce tie bloky distortion efektu, ktoré sú realizované číslicovými filtermi. Tieto filtre majú v podstate rovnakú štruktúru. Na začiatku sú zadané hodnoty jednotlivých súčiastok, následne sú vypočítané koeficienty analógového filtra. Vzťahy pre tieto koeficienty boli získané prostredníctvom programu SNAP. V prípade zosilňovača s operačným zosilňovačom a tónovej clony distortion efektu sú hodnoty ovládacích prvkov zadané ešte

pred volaním skriptu. V skripte sú koeficienty analógovej predlohy prepočítané na koeficienty číslicového filtra pomocou funkcie `bilinear`. Následne sú koeficienty číslicového filtra predané ako parametre funkcie `filter` spolu so signálom, ktorý má byť filtrovaný. Na záver každého skriptu je pridaný kód, ktorý slúži na vykreslenie prevodovej charakteristiky konkrétneho filtra.

4.2 Implementácia v jazyku C++

Po odladení algoritmu v prostredí Matlab nasledovala implementácia v jazyku C++. Vytvoriť hneď simuláciu celého distortion efektu, ktorý by pracoval v reálnom čase by bolo príliš obtiažne. Preto bolo potrebné implementovať jednotlivé bloky po častiach. Cieľom bolo naprogramovať konzolovú aplikáciu, ktorá bude simulovať chovanie jednotlivých jadier distortion efektu. Až tak tento program prispôbiť na prácu v reálnom čase. Následne simulované jadrá, funkčné a dostatočne rýchle, implementovať do VST šablóny a naprogramovať číslicové filtre.

4.2.1 Trieda matica

Jedna sa o triedu na prácu s maticami. Je vhodná aj na prácu s vektormi, nakoľko sú to špeciálne prípady matíc. V tejto triede sú matice vnímané ako jednorozmerné pole, ktorého dĺžka je určená súčinom počtu riadkov a stĺpcov matice. Je to efektívnejšie ako dvojrozmerné pole a hlavne je to rýchlejšie a teda vhodné pre prácu v reálnom čase. Triedu je možné v programe deklarovať dvoma spôsobmi, konkrétne

```
matrica X; alebo matrica X(row,col);
```

Trieda teda obsahuje dva konštruktory. Prvý bez parametrov, ktorý sa využíva na zavedenie matice v programe, ak ešte nie je známe, aké rozmery má mať. Druhý konštruktor má dva parametre, konkrétne počet riadkov a počet stĺpcov. V tomto konštruktore sa volá funkcia `alloc`, ktorá dynamicky alokuje miesto. Deštruktor obsahuje jednoduchú podmienku, na základe ktorej je možné zistiť, ktorým z konštruktorov bola matica inicializovaná, a teda či je potrebné dealokovať pamäťové miesto. Ak áno, je zavolaná funkcia `dealloc` na uvoľnenie pamäti. V tejto triede nie sú obsiahnuté všetky operácie s maticami, je to trieda, ktorá má slúžiť pri simulácii pomocou SSMN metódy a bola vytvorená čiste k tomuto účelu. Jednoduché operácie, ako súčet, rozdiel, či skopírovanie jednej matice do druhej sa dá jednoducho vyriešiť pomocou cyklu `for` s počtom opakovaní rovným dĺžke jednorozmerného poľa, ktorá je známa. Bolo však potrebné vytvoriť funkcie na násobenie dvoch matíc, výpočet inverznej matice a vytvorenie jednotkovej matice. Počas vývoja programu som zistil, že funkcie s návratovými hodnotami značne spomaľujú chod programu,

rovnako tak preťaženia operátorov. Na základe týchto poznatkov som sa rozhodol vytvoriť funkcie, ktoré budú volané prostredníctvom jednotlivých matíc a nebudú to funkcie s návratovou hodnotou.

Funkcia nasob

Táto funkcia slúži na prenášobenie dvoch matíc odovzdaných v parametroch funkcie a uloženie výsledku do matice, prostredníctvom ktorej bola funkcia volaná, napríklad `p.nasob(P1,x);`. Na začiatku tejto funkcie je potrebné výslednú maticu naplniť nulami, aby neovplyvnila výsledok súčinu, nakoľko sa pri výpočtoch medzi výsledky pripočítavajú k jednotlivým prvkom. Jedná sa o regulérny maticový súčin. Je preto potrebné využiť tri cykly s pevným počtom opakovaní. Prvé dva slúžia na určenie pozície vo výslednej matici, tretí slúži na pohyb po riadkoch prvej matice a stĺpcoch druhej matice, ktorých prvky su prenášobované. V tejto funkcii predpokladám, že rozmery matíc sú také, aké by mali pri súčine matíc byť, teda že výsledná matica má počet riadkov prvej matice a počet stĺpcov druhej matice a že počet stĺpcov prvej matice je rovnaký ako počet riadkov druhej matice.

Funkcia inv

Funkcia na výpočet inverznej matice je postavená na Gaussovej eliminačnej metóde, ktorá je v tejto funkcii rozdelená na niekoľko krokov. Upravujú sa matice odovzdané ako vstupné parametre. Funkcia je opäť volaná prostredníctvom matice, kde má byť uložená výsledná matica. Príkaz je `jac.inv(Ij1,Ij2);`, pričom prvá z matíc je tá, ku ktorej chceme vypočítať inverznú maticu a druhá je jednotková matica. Na tvorbu jednotkovej matice je vhodná funkcia `eye()`, ktorá je súčasťou triedy `matrica`. Následne sú na obe matice aplikované riadkové úpravy s cieľom vytvoriť jednotky na hlavnej diagonále prvej matice a nuly pod hlavnou diagonálou. V ďalšom kroku nastáva vytvorenie núl aj nad hlavnou diagonálou. Podarí sa nám teda preklopenie matíc. Prvá je jednotková a druhá je inverzná k pôvodnej matici. Následne stačí cez jeden `for` cyklus skopírovať druhú maticu do výslednej matice. Pokiaľ nenastane v programe chyba, prvá matica bude vždy regulárna, teda bude k nej existovať inverzná matica.

4.2.2 Trieda core

Jedná sa o triedu na prácu so samotnými nelineárnymi jadrami distortion efektu. Obsahuje viacero premenných. V prvom rade obsahuje jedno celočíselné číslo, ktoré určuje rád obvodu. Následne všetky potrebné matice, ktorými sa popisuje obvod pomocou SSMN metódy. Ďalej obsahuje niekoľko pomocných matíc, ktoré slúžia na

urýchlenie výpočtov pri násobení, alebo pri výpočtoch inverzných matíc. Vektory obsiahnuté v metóde SSMN sú reprezentované ako matice s jedným riadkom, alebo jedným stĺpcom. Trieda ďalej obsahuje dva ukazovatele na funkcie. Tieto funkcie sa zadávajú ako parametre v konštruktore triedy a jedná sa o funkciu na naplnenie matíc popisujúcich konkrétny obvod a funkciu obsahujúcu nelineárne rovnice, ktoré sú v algoritme riešené numericky. Túto triedu je v programe možné deklarovať nasledovne `core d(3,GetSampleRate(),var3,num3);`. Zadávané parametre predstavujú rád simulovaného obvodu, vzorkovaciu frekvenciu, s ktorou má simulácia pracovať, názov funkcie s premennými popisujúcimi systém a názov funkcie, ktorá obsahuje nelineárnu sústavu rovníc. Pri deklarácii sú v podstate vykonané prvé tri operácie z vývojového diagramu SSMN metódy (obr. 4.1). V deštruktore tejto triedy je postupne volaný deštruktor triedy matica pre jednotlivé matice.

Funkcia newton

Funkcia `newton` slúži na numerické riešenie nelineárnej sústavy rovníc. Funkcia má päť parametrov. V programe je volaná zápisom `newton(pt2f,in_1,K,p,in);`. Prvý parameter je ukazovateľ na nelineárnu funkciu, ktorý má byť numericky riešená, ďalšie tri parametre slúžia na určenie hodnoty, ktorá je považovaná za počiatočný odhad riešenia a posledný parameter určuje, do ktorej premennej sa výsledok uloží. Na numerické riešenie je využívaná Newtonova metóda, pre ktorú platí vzťah

$$x_n = x_{n-1} + J_F(x_{n-1})^{-1} \cdot F(x_{n-1}), \quad (4.1)$$

kde x_n je koreň nelineárnej sústavy rovníc a x_{n-1} je odhad koreňa, od ktorého začína Newtonova metóda aproximovať ku koreňu. Počiatočný odhad koreňa musí byť zvolený dostatočne presne, aby bolo možné použiť Newtonovu metódu. Funkčná hodnota pre x_{n-1} je $F(x_{n-1})$ a $J_F(x_{n-1})^{-1}$ je inverzná Jacobiho matica pre funkciu $F(x_{n-1})$. Jacobiho matica je tvorená parciálnymi deriváciami funkcie. Z tohoto dôvodu sa ako prvé vo funkcii počítajú parciálne derivácie, ktoré sú postupne vpísané do Jacobiho matice, následne je vypočítaná inverzná matica. Tieto operácie sa vykonávajú pri každom volaní funkcie len raz. Následne je inverzná Jacobiho matica dosadená do vzorca (4.1). Použitím cyklu `while` sa prepočet x_n prevádza pokiaľ nie je dosiahnutý dostatočne presný výsledok, čo určuje chyba $\varepsilon = x_n - x_{n-1}$. Výsledok sa považuje za dostatočne presný, ak $\varepsilon \leq \pm 1 \cdot 10^{-6}$. Aby sa program nezacyklil v prípade, kedy by nebolo možné dosiahnuť dostatočnú presnosť, je cyklus obmedzený maximálnym počtom iterácií.

Funkcia clipper

Funkcia `clipper` je volaná prostredníctvom inštancie typu `core` napríklad ako `d.clipper(input,output,frames);`. Vstupné parametre tejto funkcie sú ukazovatele na pole vstupného a výstupného signálu a rovnako aj ich dĺžka. Funkcia vykonáva časť operácií z vývojového diagramu (obr. 4.1), konkrétne zacyklenú časť, od načítania vstupu až do konca. Zacyklenie je vo funkcii zabezpečené pomocou cyklu `for`. Počet opakovaní určuje dĺžka spracovávaného signálu. V tejto funkcii sa volá funkcia `newton`. Na konci cyklu je výsledok skopírovaný do poľa výstupného signálu.

4.2.3 Knižnica `fcie.h`

Táto knižnica obsahuje všetky potrebné funkcie na popis jednotlivých systémov, ktoré majú byť simulované. Ku každému obvodu patria dve funkcie, vždy jedna s názvom `varx` a druhá s názvom `numx`, pričom `x` predstavuje rád simulovaného obvodu a v programe je tento znak nahradený číslom.

Funkcie `var`

Funkcia bez návratovej hodnoty, ako jediný parameter sa odovzdáva dátová premenná typu `core`, prostredníctvom ktorej dokážeme pristupovať k jednotlivým matriciam. Tieto matice sú vo funkcii postupne naplňané. Na záver sú vypočítané pomocné matice, ktoré sa ďalej v programe nemenia. Táto funkcia sa volá len raz a to z konštruktora triedy `core` príkazom `pt2v(*this);`, nakoľko funkcia je reprezentovaná ukazovateľom.

Funkcie `num`

Volanie tejto funkcie sa vykonáva behom programu pravidelne príkazom napríklad `pt2f(pomh,Ka,pe,der);`. Obsahuje totiž nelineárnu sústavu rovníc. Táto sústava popisuje nelineárnu časť simulovaného obvodu. Názov tejto funkcie sa uloží do ukazovateľa funkcie v triede `core`. Tento ukazovateľ sa odovzdáva ako parameter funkcie `newton`. Funkcia nemá návratovú hodnotu a má tri vstupné parametre.

4.3 Implementácia ako VST plug-in modul

Aby bolo možné simulovaný efekt používať na spracovávanie signálov v reálnom čase, je potrebné zaobaliť ho do zásuvného modulu, s ktorým dokáže pracovať množstvo hosťovských aplikácií. K tomuto účelu som využil VST šablónu od firmy Steinberg, ktorá bola prispôbena doktorom Schimmelom z Ústavu teleinformatiky

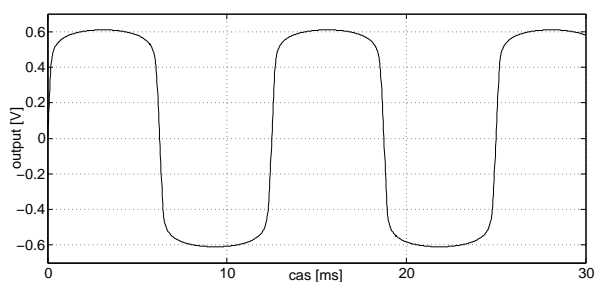
a komunikačných technológií VUT v Brne. Do tejto šablóny bol odladený program postupne vpisovaný. Na orientáciu v šablóne slúži literatúra [7].

4.4 Výpočtová náročnosť

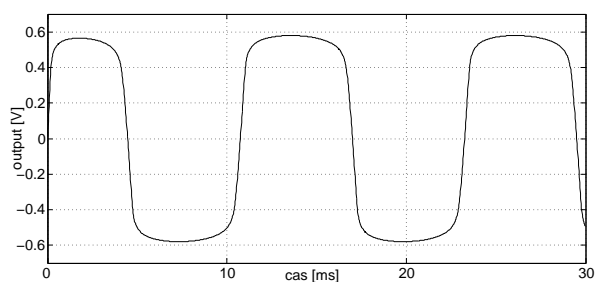
Optimalizovaný program pri spracovaní harmonického ustáleného signálu dĺžky 16000 vzoriek pri vzorkovacej frekvencii 48kHz vykazuje pri simulácii dvoch jadier tretieho a druhého stupňa oneskorenie približne 310ms pri spracovaní jedného zvukového kanálu. Nakoľko signál tejto dĺžky pri vzorkovacej frekvencii 48kHz trvá 333ms , je možné prehlásiť, že program je prispôsobený na prácu v reálnom čase. Výpočtová náročnosť číslicových filtrov je zanedbateľná oproti výpočtovej náročnosti metódy SSMN.

5 VÝSLEDKY

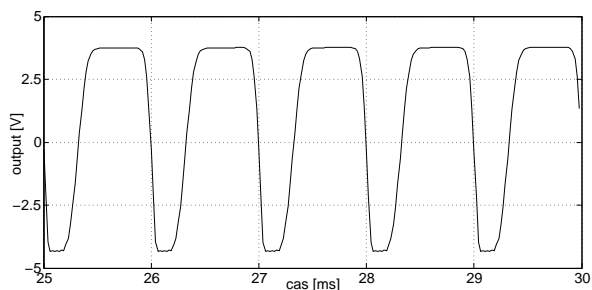
Na (obr. 5.1, 5.2, 5.3) môžeme vidieť odozvy simulácie v prostredí Matlab pomocou SSMN metódy pre jednotlivé zapojenia diódového orezávača a tranzistorového zosilňovača. Vo všetkých prípadoch je využitý jeden algoritmus, čím bola overená hromadnosť algoritmu. V prvých dvoch prípadoch bol ako vstupný signál použitý sínusový signál s periódou $4,5V$ a frekvenciou $80Hz$. V treťom prípade to bol sínusový signál s amplitúdou $0,2V$ a frekvenciou $1kHz$.



Obr. 5.1: Odozva diódového orezávača s RC filtrom - SSMN metóda

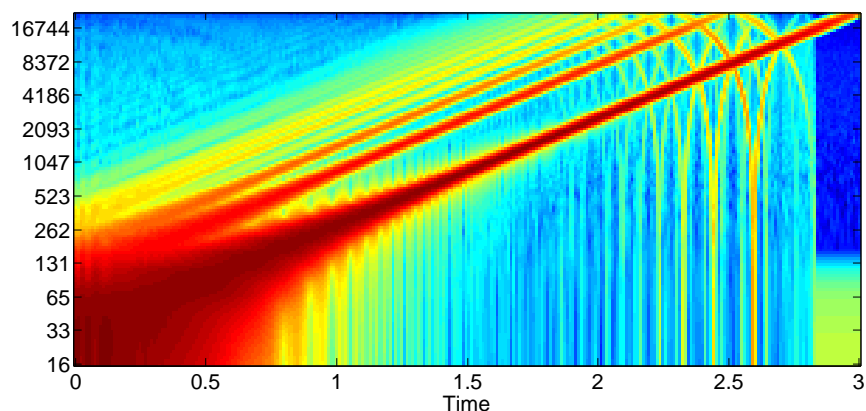


Obr. 5.2: Odozva diódového orezávača s dvoma kondenzátormi - SSMN metóda

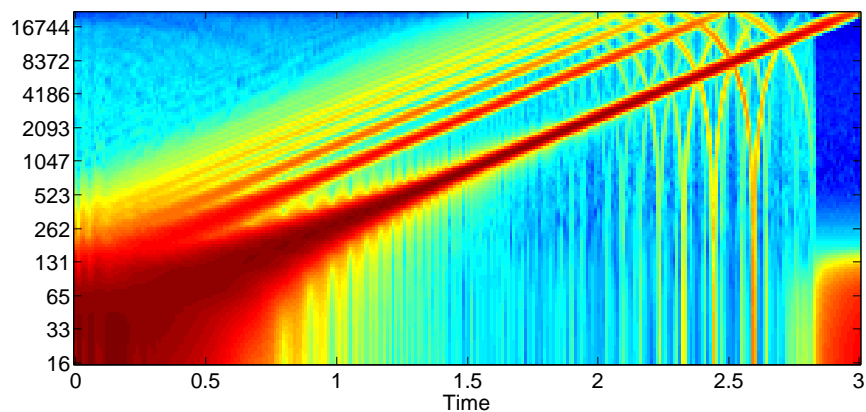


Obr. 5.3: Odozva tranzistorového zosilňovača so spätnou väzbou - SSMN metóda

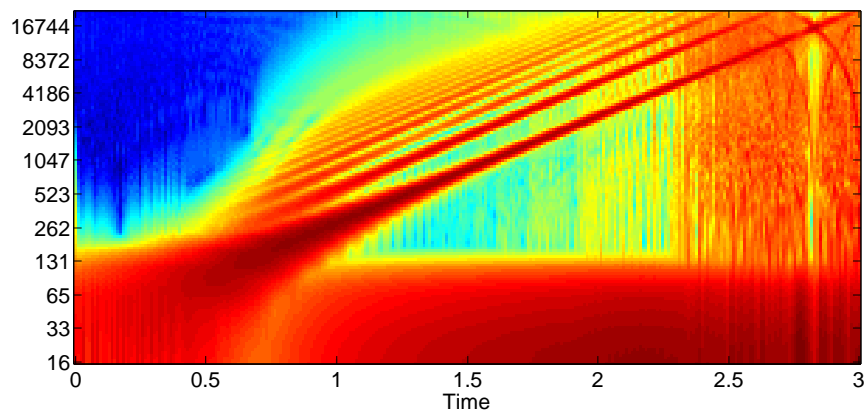
Na (obr. 5.4, 5.5, 5.6) sú zobrazené spektrogramy výstupných signálov simulovaných obvodov pri použití rozmietavého sínusového signálu ako vstupného signálu do systému. V prípade simulácie tranzistorového zosilňovača sa prejavuje aliasing viac ako v obvodoch s diódovým orezávačom.



Obr. 5.4: Log. spektrogram signálu „Sine sweep“ pre diódový orezávač s RC filtrom

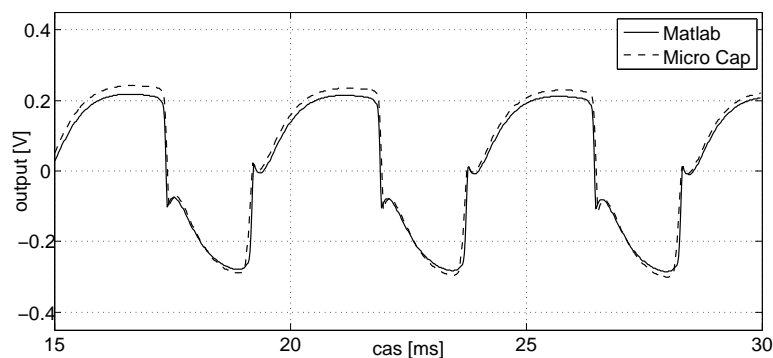


Obr. 5.5: Log. spektrogram signálu „Sine sweep“ pre diódový orezávač s dvoma kondenzátormi



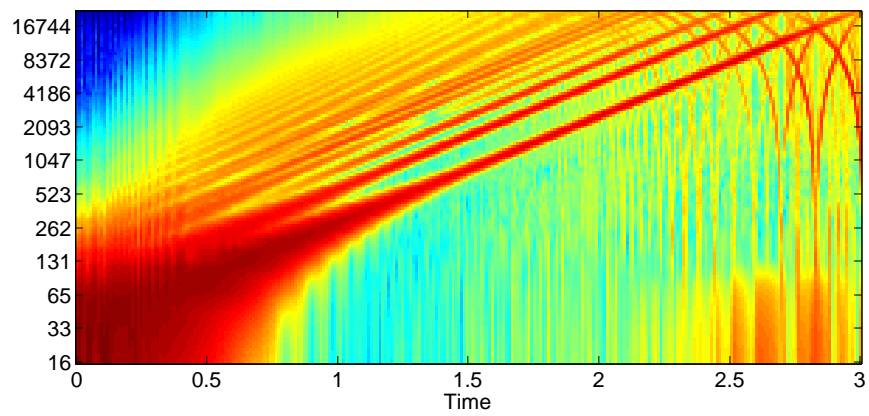
Obr. 5.6: Log. spektrogram signálu „Sine sweep“ pre tranzistorový zosilňovač so spätnou väzbou

Po nasimulovaní celého distortion efektu v prostredí Matlab bola spravená simulácia aj v simulačnom programe Micro Cap. Na (obr. 5.7) môžeme vidieť porovnanie výstupného signálu distortion efektu simulovaného v prostredí Matlab a v simulačnom programe Micro Cap. Ako vstupný signál bol použitý sínusový signál s amplitúdou $0,3V$ a frekvenciou $220Hz$. Výsledky sú porovnateľné, nakoľko sa vo vysoko nelineárnych častiach takmer prekrývajú.

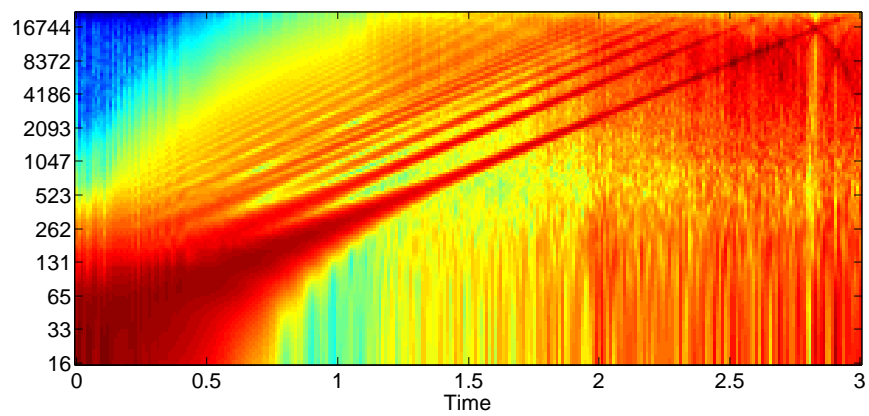


Obr. 5.7: Porovnanie simulovaných efektov v Matlabe a Micro Cape

Na (obr. 5.8, 5.9) sú znázornené spektrogramy výstupných signálov z oboch simulácií. V spektrograme pri simulácii programom Micro Cap je zjavne menší aliasing, čo bolo spôsobené vyššou vzorkovacou frekvenciou ($100kHz$) oproti vzorkovacej frekvencii v Matlabe ($48kHz$).



Obr. 5.8: Log. spektrogram signálu „Sine sweep“ distortion efektu - simulácia Micro Cap



Obr. 5.9: Log. spektrogram signálu „Sine sweep“ distortion efektu - simulácia Matlab

6 ZÁVER

Po preštudovaní rôznych metód simulácie gitarových hudobných efektov bola za metódu využívanú v tejto práci vybraná metóda stavového priestoru so statickou nelinearitou (SSMN).

Nasledovalo intenzívnejšie preštudovanie a pochopenie tejto metódy, zostavenie vývojového diagramu z výpočtového algoritmu tejto metódy, ktorý je uvedený v tejto práci a následná implementácia algoritmu v prostredí Matlab pre vybrané obvody.

Ďalším krokom bol výber gitarového hudobného efektu. Vybraný bol efekt distortion, z dôvodu rozšírenia tohoto efektu v hudobnom priemysle. Po analýze nelineárnych blokov a nájdení všetkých potrebných premenných bolo možné tieto bloky simulovať. Nasledovala realizácia číslicových filtrov, ktoré predstavujú zvyšné bloky distortion efektu. Po simulácii celého efektu boli výsledky porovnané s výsledkami zo simulačného programu Micro Cap. Výsledky simulácií pre jednotlivé nelineárne bloky, ako aj pre celý distortion efekt sú uvedené v kapitole (5) tejto práce.

Po simulácii v prostredí Matlab bolo potrebné implementovať odladený algoritmus v jazyku C++. Pri tvorbe programu bolo potrebné myslieť nato, aby bolo možné ho prispôsobiť pre prácu v reálnom čase. Preto sú matice v programe vnímané ako jednorozmerná pole a nie je využívané preťažovanie operátorov, ani funkcie s návratovými hodnotami. Výsledkom práce je efekt distortion simulovaný v prostredí Matlab, rovnako tak prispôsobený algoritmus na simuláciu nelineárnych obvodov SSMN metódou v konzolovej aplikácii jazyka C++ a funkčný VST plug-in modul simulujúci distortion efekt.

Algoritmus sa podarilo prispôsobiť na prácu v reálnom čase a výsledná simulácia distortion efektu bola testovaná na extérnej zvukovej karte M-Audio Fast Track. Výpočtová náročnosť simulácie jedného jadra je daná ako $O(N^2 + NM + NP)$, kde N je počet stavov, M je počet vstupov a P je počet výstupov systému, ak je numerické riešenie nelineárnej rovnice vyčíslené behom jednej iterácie [11]. Táto metóda nie je vhodná na simuláciu hudobných efektov, ktoré majú silne nelineárne bloky, alebo viac ako dva nelineárne bloky, nakoľko by výpočtová náročnosť bola príliš vysoká.

LITERATÚRA

- [1] FETTWEIS, A.: *Wave digital filters: Theory and practice*, Proc. IEEE, vol. 74, no. 2, pp. 270 – 327, Feb. 1986.
- [2] MAČÁK, J.; SCHIMMEL, J.: *Real-Time Guitar Tube Amplifier Simulation using Approximation of Differential Equations*. In Proc. Digital Audio Effects DAFx10, Graz, Austria, Sep. 6–10, 2010.
- [3] NOVAK, A.; SIMON, L.; LOTTON, P.; KADLEC, F.: *Modeling of nonlinear audio systems using swept-sine signals: application to audio effects* In Proc. Digital Audio Effects DAFx09, Como, Italy, Sep. 1–4, 2009, s. 1–6.
- [4] PAKARINEN, J.; KARJALAINEN, M.: *Wave digital simulation of a vacuum-tube amplifier*. In Proc. Intl. Conf. on Acoustics, Speech, and Signal Proc., Toulouse, France, May 15–19, 2006, s. 153–156.
- [5] ROLAND CORP.: *Boss DS-1 Service Notes*, Dec. 26 1980.
- [6] SERAFINY, T.; BARBATI, S.: *A Perceptual Approach on Clipping and Saturation*, 2002, [online], Dostupné z <<http://www.simulanalog.org/clip.pdf>>.
- [7] SCHIMMEL, J.: *Šablóna VST plug-in modulu*, Vysoké učení technické v Brně, 4. 5. 2008
- [8] YEH, T. D.: *Digital Implementation of Musical Distortion Circuits by Analysis and Simulation*, Disertační práce, Stanford University, June 2009.
- [9] YEH, T. D.; ABEL, J. S.; SMITH, J. O.: *Simulation of the diode limiter in guitar distortion circuits by numerical solution of ordinary differential equations* in Proc. Digital Audio Effects (DAFx-07), Bordeaux, France, Sep. 10–15, 2007, pp. 197–204.
- [10] YEH, T. D.; ABEL, J. S.; SMITH, J. O.: *Simplified, Physically-Informed Models of Distortion and Overdrive Guitar Effects Pedal* In Proc. Digital Audio Effects (DAFx-07), Bordeaux, France, Sep. 10–15, 2007, s. 189–196.
- [11] YEH, T. D.; SMITH, J. O.: *Simulating Guitar Distortion Circuits Using Wave Digital and Nonlinear State-Space Formulations*. In Proc. Digital Audio Effects (DAFx-08), Espoo, Finland, Sept. 1–4, 2008, s. 19–26.
- [12] ZÖLZER, U.: *DAFX - Digital Audio Effects*, J. Wiley & Sons, Ltd, prvé vydanie, 2002. ISBN 0-471-49078-4, 533 s.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

SSMN	metóda stavového priestoru so statickou nelinearitou - state space memoryless nonlinearity
$\mathbf{A} - \mathbf{F}$	lineárne kombinácie stavov, vstupov a príspevkov od nelineárnych častí systému
$\mathbf{A}_o - \mathbf{C}_o$	lineárne kombinácie stavov, vstupov a príspevkov od nelineárnych častí systému pre zostavenie výstupu
C	kapacita
f_{vz}	vzorkovacia frekvencia
G	vodivosť
i	príspevok od nelineárnej časti systému
\mathbf{I}	jednotková matica
I_c	kolektorový prúd
I_b	bázový prúd
I_d	prúd diódou
$I_d(U)$	prúd diódovým orezávačom
I_s	saturačný prúd
p	lineárna kombinácia stavu a vstupu do systému
R	odpor
T	perióda vzorkovacej frekvencie
u	vstup do systému
U	napätie na diódovom orezávači
U_{cb}	napätie na diódovom orezávači
U_{bc}	napätie na diódovom orezávači
U_i	napätie na vstupe RC filtra
U_o	výstupné napätie

U_t	termálne napätie
\boldsymbol{v}	vektor napätí na nelineárnych prvkoch
VST	Vistual Studio Technology - štandard používaný pri digitálnych audio systémoch
\boldsymbol{x}	vektor predstavujúci pamäť systému (napätie kondenzátorov, prúd cievok)